

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Bakalářská práce

2011

Petra Sedláčková

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Systém pro generování modelu databáze
System for Database Model Generation

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně.

Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

Dne: 16. 7. 2011

Petra Sedláčková

Podpis

Poděkování

Ráda bych poděkovala vedoucí mé bakalářské práce, paní Ing. Emilii Šeptákové, za odbornou pomoc a čas strávený při konzultacích.

Abstrakt

Databáze v dnešní době používá většina webových aplikací, zejména z důvodu nutnosti ukládat používaná data. Cílem této bakalářské práce je nejprve srovnání již vytvořených systémů pro správu databází a poté navrzení, tvorba a naprogramování nového systému, který umožní zadání přihlašovacích údajů k SQL serveru. Práce se zabývá tvorbou výpisů seznamu databází, tabulek z jedné databáze, atributů z vybrané tabulky, vazby, ER diagram jedné databáze, datového slovníku, a lineárním zápisem entit vybrané databáze. K takto vytvořenému systému byla navržena programátorská a uživatelská příručka. V závěru je věnována pozornost srovnání takto naprogramovaného systému se systémy již existujícími. Nově vytvořený systém může sloužit jako dokumentace k systému již vytvořenému.

Klíčová slova

Databáze, PHP, MySQL, SQL, HTML, ERD.

Abstract

Currently databases are used by most web application, which results from the need to keep and save used data. The aim of the bachelor work is to compare still existing systems for databases management and then design, develop and program a new system, in which login to SQL server is applicable. The second part focuses on development of the statement of the list of databases, tablets from one database, attributes from a selected tablet, links, ER diagram, data vocabulary, and linear report of entities of a selected database. For such a developed system the applicable programmer and user guide has been designed. The final part deals with the comparison of such developed programmed system to still existing ones. The newly developed system may serve as a documentation tool for the systems already in existence.

Key words

Database, PHP, MySQL, SQL, HTML, ERD.

Seznam použitých symbolů a zkratk

| | |
|------|---|
| PHP | Hypertextový preprocesor, původně Personal Home Page |
| SQL | Structured Query Language, strukturovaný dotazovací jazyk |
| ERD | Entity Relationship Diagram |
| HTML | HyperText Markup Language |
| IO | Integritní Omezení |

Seznam tabulek

| | | |
|------------|--|----|
| Tabulka 1. | Srovnání systémů..... | 13 |
| Tabulka 2. | Část datového slovníku databáze obchod | 15 |
| Tabulka 3. | Tabulka schemata – seznam databází v SŘBD..... | 18 |
| Tabulka 4. | Tabulka tables..... | 19 |
| Tabulka 5. | Tabulka columns..... | 20 |

Seznam obrázků

| | | |
|-------------|---|----|
| Obrázek 1. | Základní obrazovka | 3 |
| Obrázek 2. | Vytvoření nové databáze | 3 |
| Obrázek 3. | Formuláře databáze knihovna | 4 |
| Obrázek 4. | Náhled k vytištění | 5 |
| Obrázek 5. | Vytvoření nové tabulky | 5 |
| Obrázek 6. | Formuláře pro SQL dotaz v databázi knihovna | 6 |
| Obrázek 7. | Struktura tabulky exemplář z databáze knihovna | 6 |
| Obrázek 8. | Údržba tabulky | 7 |
| Obrázek 9. | Základní obrazovka Workbench 5.0 | 8 |
| Obrázek 10. | Okno pro SQL skript | 9 |
| Obrázek 11. | Vykreslení ER diagramu | 9 |
| Obrázek 12. | Informace o serveru | 10 |
| Obrázek 13. | Výpis tabulky vypůjčka z databáze knihovna | 11 |
| Obrázek 14. | MySQL Query Browser nápověda | 12 |
| Obrázek 15. | ER diagram databáze obchod | 15 |
| Obrázek 16. | ER diagram | 17 |
| Obrázek 17. | Připojení k databázi MySQL | 23 |
| Obrázek 18. | Hlavička tabulky pro výpis atributů | 23 |
| Obrázek 19. | Kód pro načtení hodnot | 24 |
| Obrázek 20. | Kód pro vypsání podtrženého textu a kurzívou | 24 |
| Obrázek 21. | Příklad hlavičky HTML | 25 |
| Obrázek 22. | Část kódu CSS | 26 |
| Obrázek 23. | Zdrojový kód databáze knihovna pokl.mp | 27 |
| Obrázek 24. | Erd.tex | 27 |
| Obrázek 25. | Výpis seznamu databází | 28 |
| Obrázek 26. | Formulář pro přihlášení | 28 |
| Obrázek 27. | Výpis tabulek databáze knihovna | 29 |
| Obrázek 28. | Výpis atributů z tabulky exemplar a tabulky závislé | 29 |
| Obrázek 29. | Lineární zápis typu entit a vazeb databáze knihovna | 30 |

| | | |
|-------------|-----------------------------|----|
| Obrázek 30. | Část datového slovníku..... | 30 |
| Obrázek 31. | Vykreslení ER diagramu..... | 31 |

Obsah

| | | |
|-------|---|----|
| 1 | Úvod | 1 |
| 1.1 | Historie | 1 |
| 2 | Porovnání | 2 |
| 2.1 | MySQL | 2 |
| 2.2 | PhpMyAdmin | 3 |
| 2.3 | MySQL Workbench | 8 |
| 2.4 | MySQL Administrator | 10 |
| 3 | Navrhovaný systém | 14 |
| 3.1 | Funkce | 14 |
| 4 | Datová analýza | 16 |
| 4.1 | Lineární zápis | 16 |
| 4.2 | Vazby | 17 |
| 4.3 | ER diagram | 17 |
| 4.4 | Datový slovník + integritní omezení | 18 |
| 5 | Implementace systému | 21 |
| 5.1 | PHP | 21 |
| 5.1.1 | Historie PHP | 22 |
| 5.1.2 | Psaní PHP skriptů | 22 |
| 5.1.3 | Ukázky PHP kódu | 22 |
| 5.2 | HTML | 24 |
| 5.2.1 | Syntaxe jazyka HTML | 24 |
| 5.3 | CSS | 25 |
| 5.4 | Generování ER diagramu do obrázku | 26 |
| 6 | Provoz | 28 |
| 6.1 | Vzhled systému | 28 |
| 7 | Závěr | 32 |

1 Úvod

Tato bakalářská práce se zabývá nejprve porovnáním rozdílů již existujících systémů pro správu databáze. Dále návrhem vhodných funkcí, atributů, SQL dotazů, datového slovníku a vytvoření vlastního systému pro generování ER diagramu, lineárního zápisu typu entit a vazeb pro vybranou databázi. K tomuto systému sepsat uživatelskou a programátorskou příručku. A na závěr porovnat vytvořený systém s již existujícími systémy.

1.1 Historie

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. Součástí databáze jsou i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software nazýváme systém řízení báze dat (SŘBD).

Předchůdcem databází byly papírové kartotéky. Data byla uspořádána podle různých kritérií a snadno se vytvářely nové položky. Veškerou manipulaci s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází.

Dalším krokem bylo zpracování dat strojem. Paměťovým médiem byl děrný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích.

V roce 1959 se konala konference zástupců firem, uživatelů a amerického ministerstva obrany, jejímž závěrem byl požadavek na univerzální databázový jazyk. Výsledkem byla o rok později na konferenci CODASYL publikovaná první verze jazyka COBOL, který byl po mnoho dalších let nejrozšířenějším jazykem pro hromadné zpracování dat.

O 7 let začaly vznikat první síťové SŘBD. Jedním z prvních průkopníků databází byl Charles Bachman.

V roce 1971 se objevily pojmy jako schéma databáze, jazyk pro definici schématu, subschéma a podobně. Byla zde popsána celá architektura síťového databázového systému. Začínají první relační databáze, které pohlíží na data jako na tabulky. Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL.

V 90. letech 20. století se začínaly objevovat první objektově orientované databáze, jejichž filozofie byla přebírána z objektově orientovaných jazyků. Tyto databáze měly podle předpokladů vytlačit relační systémy. Původní předpoklady se však nenaplnily a vznikla kompromisní objektově-relační technologie [1].

2 Porovnání

K porovnávání již existujících systémů byly vybrány systémy phpMyAdmin, MySQL Workbench a MySQL Administrator. Tyto systémy pracují na databázi MySQL.

2.1 MySQL

MySQL (zkratka z angl. My Structured Query Language = *systém pro řízení databází*) je švédský databázový server založený na jazyce SQL. Je to program, který je k dispozici zdarma. K dalším výhodám MySQL patří podpora všech hlavních platforem, vysoký výkon i rychlost a vynikající kompatibilita s jinými systémy, zejména se serverovým programem Apache a skriptováním PHP. MySQL je také díky své jednoduchosti snadno pochopitelný. Díky těmto vlastnostem se MySQL prosadil jako univerzální řešení používané na většině internetových projektů a je automaticky dostupná téměř na všech typech webhostingu.

Nevýhody MySQL vychází z jejích výhod. Nepodporuje složitější programátorské konstrukce a nemá dostatečný výkon v opravdu náročných (zatěžovaných) webových aplikacích. Tehdy se používají konkurenční databáze, například PostgreSQL nebo Oracle. Přesto je však třeba říci, že MySQL vyhoví ve většině případů.

Do MySQL lze ukládat různá data (texty, obrázky atd.), s nimiž lze dále jednoduše pracovat (třídít, řadit, filtrovat apod.). Nejčastěji se MySQL používá ve spojení s jazykem PHP, které umožňuje přístup k uloženým datům.

Každá databáze v MySQL obsahuje *tabulky*, každá tabulka má *sloupce a řádky* – v každém řádku jsou záznamy předem určeného typu.

Databáze MySQL je jeden z prvních hojně rozšířených systémů. Práce s tímto systémem se dá využít v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo .NET.

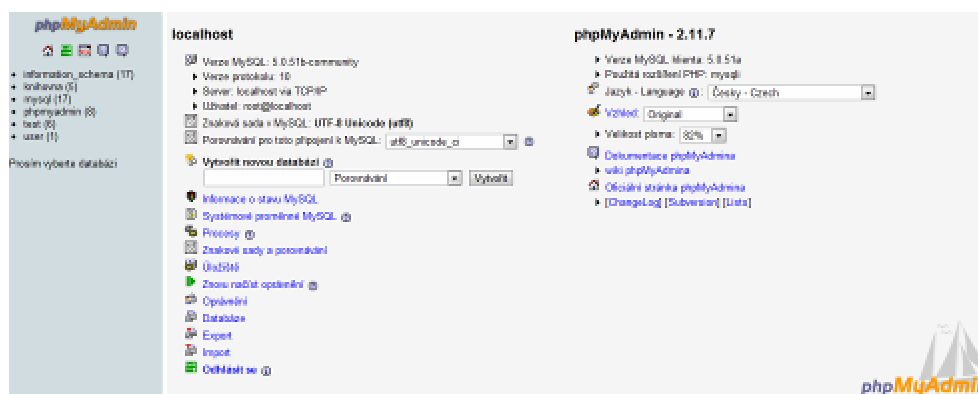
Pro jednoduchou správu MySQL databází se používá nástroj phpMyAdmin. PhpMyAdmin je Open Source program napsaný v PHP, který umožňuje zálohování, vytváření tabulek, vkládání, editaci a mazání záznamů v tabulkách, vytváření databází apod. PhpMyAdmin je pokročilý nástroj pro kompletní správu MySQL systému přes webové rozhraní.

2.2 PhpMyAdmin

PhpMyAdmin je webová aplikace napsaná v jazyce PHP. Je určena pro správu databáze MySQL přes webové rozhraní. S její pomocí lze snadno vytvořit rychle novou tabulku nebo odhalit chyby v SQL dotazech.

Přihlášení

Nejprve se přihlásím platným uživatelským jménem a heslem. Levý sloupec nám nabídne seznam dostupných databází, během práce se nemění. V pravé části obrazovky se zobrazují detaily vybrané funkce. Po přihlášení jsou zde zobrazeny vlastnosti MySQL a phpMyAdmin a odkazy na obecné funkce.



Obrázek 1. Základní obrazovka

První řádek ikon v menu tvoří rychlé odkazy na nejčastěji používané funkce. První ikona *Hlavní strana* nás přenese na počáteční stránku (obrázek 1.), která se také načte vždy ihned po přihlášení. Druhá ikona *Exit* slouží k odhlášení ze systému. Třetí ikona *SQL okno* spustí okno pro vložení SQL dotazu nebo pro import dat ze souboru. Čtvrtá a pátá ikona nám umožní přístup k dokumentaci phpMyAdmin a MySQL. Pod tímto řádkem je seznam všech databází na serveru dostupných přihlášenému uživateli. Druhá úroveň tohoto menu je seznam tabulek ve vybrané databázi.

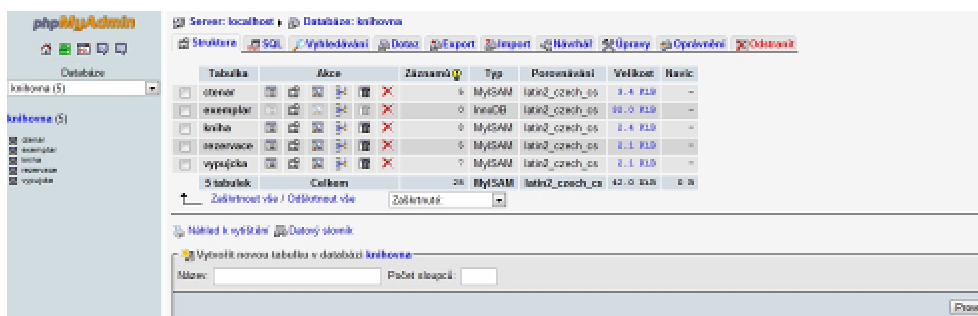
V pravé části jsou především administrační prvky týkající se celého databázového serveru. Jejich dostupnost závisí na právech právě přihlášeného uživatele. Po přihlášení jako *root* se zobrazí všechny administrační prvky. Vytvořit novou databázi může jen uživatel s příslušnými právy (obrázek 2.).



Obrázek 2. Vytvoření nové databáze

Seznam tabulek



V seznamu tabulek je pod každou uveden odkaz. Po klepnutí na něj se do pravého rámce načte formulář pro správu dané tabulky. Napravo od názvů tabulek jsou ikony pro různé akce nad tabulkou (*projít*, *struktura*, *vyhledávání*, *vložit*, *vyprázdnit*, *odstranit*) a informace o tabulkách. Odkazy na nich zobrazují výpisy obsahu tabulek, tj. `select * from tabulka`.




Obrázek 3. Formuláře databáze knihovna

Přehled všech tabulek v databázi knihovna (obrázek 3.), s odkazy *projít* (stránkovaný výpis všech vět), *struktura* (výpis atributů a jejich popis tabulky), *vyhledávání* (formulář pro vyhledávání v databázi), *vložit* (formulář pro vložení záznamů do tabulky), *vyprázdnit* (smazání vět z tabulky), *odstranit* (odstranění celé tabulky z databáze), a další údaje o správě, údaje o počtu záznamů a jejich velikosti.

Ve spodní části si můžeme zobrazit *náhled k vytištění* (obrázek 4.) a *datový slovník*.


 [Server: localhost](#) ▶  [Databáze: knihovna](#)

| Tabulka | Záznamů | Typ | Velikost | Komentáře | |
|-----------|---------|--------|----------|----------------------|-------------------------------|
| ctenar | 5 | MyISAM | 3.4 KiB | Vytvoření: | Středa 23. března 2011, 17:04 |
| | | | | Poslední změna: | Středa 23. března 2011, 18:04 |
| exemplar | 0 | InnoDB | 32.0 KiB | InnoDB free: 4096 kB | |
| | | | | Vytvoření: | Úterý 26. dubna 2011, 16:46 |
| kniha | 8 | MyISAM | 2.4 KiB | Vytvoření: | Středa 23. března 2011, 17:04 |
| | | | | Poslední změna: | Středa 23. března 2011, 18:04 |
| rezervace | 5 | MyISAM | 2.1 KiB | Vytvoření: | Středa 23. března 2011, 17:04 |
| | | | | Poslední změna: | Středa 23. března 2011, 18:04 |
| vypujcka | 7 | MyISAM | 2.1 KiB | Vytvoření: | Středa 23. března 2011, 17:04 |
| | | | | Poslední změna: | Středa 23. března 2011, 18:04 |
| 5 tabulek | 25 | -- | 42.0 KiB | | |

 [Otevřít nové okno phpMyAdmina](#)

Obrázek 4. Náhled k vytištění

Také zde máme formulář pro *vytvoření nové tabulky* v databázi knihovna (obrázek 5.).

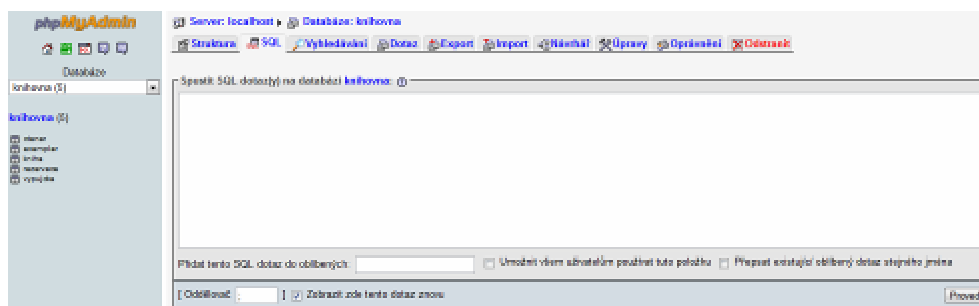
 Vytvořit novou tabulku v databázi [knihovna](#)

Název: Počet sloupců:

Obrázek 5. Vytvoření nové tabulky

Formuláře pro SQL dotaz

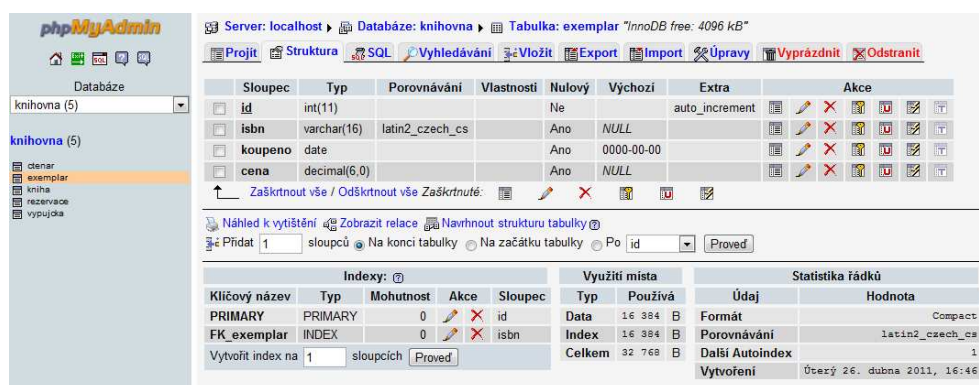
Formulář pokračuje textovým polem pro vkládání a spouštění jakéhokoli ručně napsaného SQL dotazu (obrázek 6.). Můžete také odeslat k vykonání SQL dotazy uložené v souboru na vašem počítači. Stačí vyhledat soubor v dialogovém okně a potvrdit načtení myší.



Obrázek 6. Formuláře pro SQL dotaz v databázi knihovna

Často využívanými jsou exporty: jen struktura tabulek, nebo jen obsah tabulek a nebo obě tyto možnosti. Vyberu tabulky, kterých se má export týkat. Přidej drop table znamená, že před každým create table tabulka (...) se nejprve zapíše drop table if exists tabulka. Při jejím zaškrtnutí si soubor s exportem stáhnu rovnou na lokální počítač. Jinak se objeví jako dokument v pravém rámci.

V pravé části můžu tabulku *upravit*, *vyprázdnit* nebo *odstranit*. Dále můžete vytvořit *indexy* a klíče (*Primární*, *Index* a *Unikátní*). Ve střední části máme možnost přidat sloupce do tabulky. Volby pro práci se sloupci jsou: *Změnit* (změna parametrů sloupce) a *Odstranit* (zrušení sloupce). Obojí odpovídá SQL dotazu alter table. Vše vidíme na obrázku 7.



Obrázek 7. Struktura tabulky exemplář z databáze knihovna

Výpis tabulky známe z *formuláře databáze*. Zde exportujete právě jednu tabulku. Tabulku můžeme exportovat ve formátu .csv buď ve standardu pro MS Excel nebo v takovém, jaký si sami zadáme. Můžeme ovlivnit i rozsah - které řádky se budou exportovat. Zaškrtnutím *Proved'* dojde ke stažení souboru na náš počítač, jinak se výpis zobrazí v pravém rámci.

Změnit pořadí tabulky podle

cislo (po jednom) Vzestupně Proved

Přesunout tabulku do (databáze.tabulka):

knihovna . ctenar

☒ Přidat hodnotu AUTO_INCREMENT

Proved

Parametry tabulky

Přejmenovat tabulku na ctenar

Komentář k tabulce

Úložiště MyISAM

Porovnávání latin2_czech_cs

pack_keys DEFAULT

checksum ☐

delay_key_write ☐

auto_increment 6

Proved

Kopírovat tabulku do (databáze.tabulka):

knihovna .

☐ Pouze strukturu

☒ Strukturu a data

☐ Jen data

☐ Přidat DROP TABLE

☐ Přidat hodnotu AUTO_INCREMENT

☐ Přepnout na zkopírovanou tabulku

Proved

Údržba tabulky

- [Zkontrolovat tabulku](#)
- [Analyzovat tabulku](#)
- [Opravit tabulku](#)
- [Optimalizovat tabulku](#)
- [Vyprázdnit vyrovnávací paměť pro tabulku \(FLUSH\)](#)

Obrázek 8. Údržba tabulky

Funkce *přejmenování*, *kopírování* a *přesunutí* (i do jiné databáze) si zvolíme přes jednoduché formuláře pod exportem (obrázek 8.). Pro údržbu tabulky máte k dispozici odkazy *Zkontrolovat tabulku* (v jazyce SQL: `Check table tabulka`), *Analyzovat tabulku* (Analyze table tabulka), *Opravit tabulku* (Repair table tabulka), *Optimalizovat tabulku* (Optimize table tabulka) a *Vyprázdnit vyrovnávací paměť pro tabulku*.

Změnu typu datové struktury, ve které je tabulka uložena, doporučuji pouze tomu, kdo ví, co dělá.

Pokud bych měla vlastností phpMyAdmin shrnout, pak to jsou tyto:

- vytvoření a rušení databáze
- vytvoření, kopírování, přesouvání, odstranění a změny tabulky
- přidávání, změny a odstranění sloupce v tabulkách
- spouštění SQL příkazů, včetně dávkového zpracování
- spravování klíčů a indexů tabulek
- nahrávání textových souborů do tabulek
- vytváření exportů a textových souborů z tabulek
- import dat do databáze z různých formátů
- export dat z databáze do souborů různých formátů

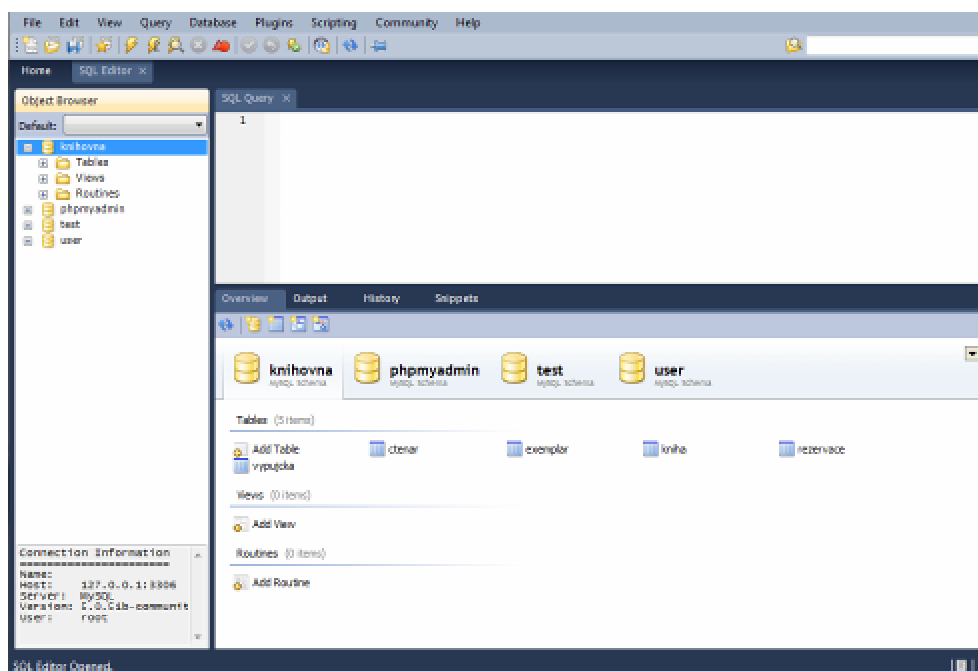
2.3 MySQL Workbench

MySQL Workbench je grafické prostředí určené pro práci s SQL serverem MySQL. Je to desktopová aplikace. MySQL Workbench nabízí mnoho funkcí.

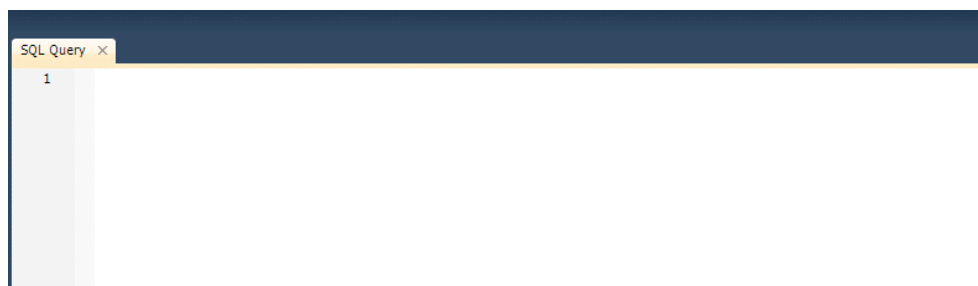
Ke klasickým funkcím patří:

- snadné vytvoření, editování a rušení databáze
- vytvoření, odstranění a změny tabulky
- vytvoření pohledů a procedur
- označení klíčů, indexy, triggerů
- editory SQL příkazů, SQL skriptů (obrázek 10.), tabulek, pohledů, uložených procedur, uložených funkcí, uložených modulů, rolí, uživatelů, vykreslení ER diagramu (obrázek 11.) a dalších
- umožňuje import a export dat

Každý editor je velmi dobře navržen a poskytuje rozsáhlé možnosti, takže administrátor nemusí nic řešit "ručně" a může opravdu téměř vše nastavit v dialogu. Velkou výhodou je, že s každým dialogem je velmi rychlá práce. Základní obrazovku vidíme na obrázku 9.



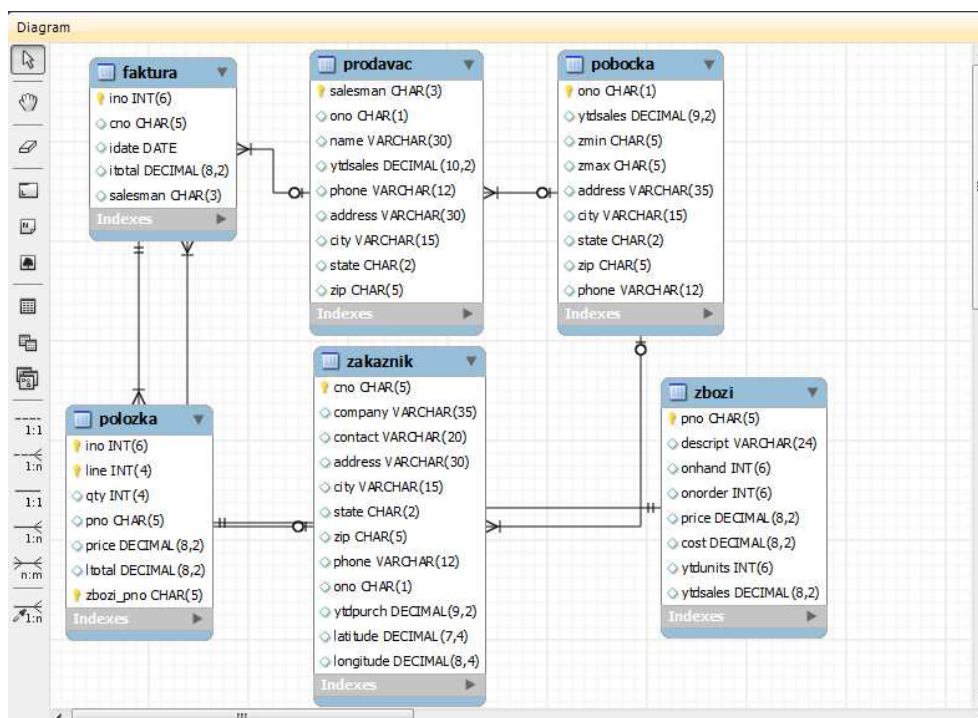
Obrázek 9. Základní obrazovka Workbench 5.0



Obrázek 10. Okno pro SQL skript

Součástí návrhu aplikací nebo analýzy existujícího systému občas potřebujeme vykreslit ER diagramy.

Ty jsou názornější než tabulky či slovní popisy a pomáhají zejména ve věcech souvisejících s relacemi (cizí klíče, apod.), viz. obrázek 11.



Obrázek 11. Vykreslení ER diagramu

Po odzkoušení těchto dvou systémů, jsem vyhodnotila, že práce s MySQL Workbench je jednodušší než s phpMyAdmin. Už žádné zdlouhavé vytváření tabulek v phpMyAdminu, nýbrž přehledné grafické prostředí. Tlačítko synchronizovat uloží celou strukturu, včetně relací do databáze [8].

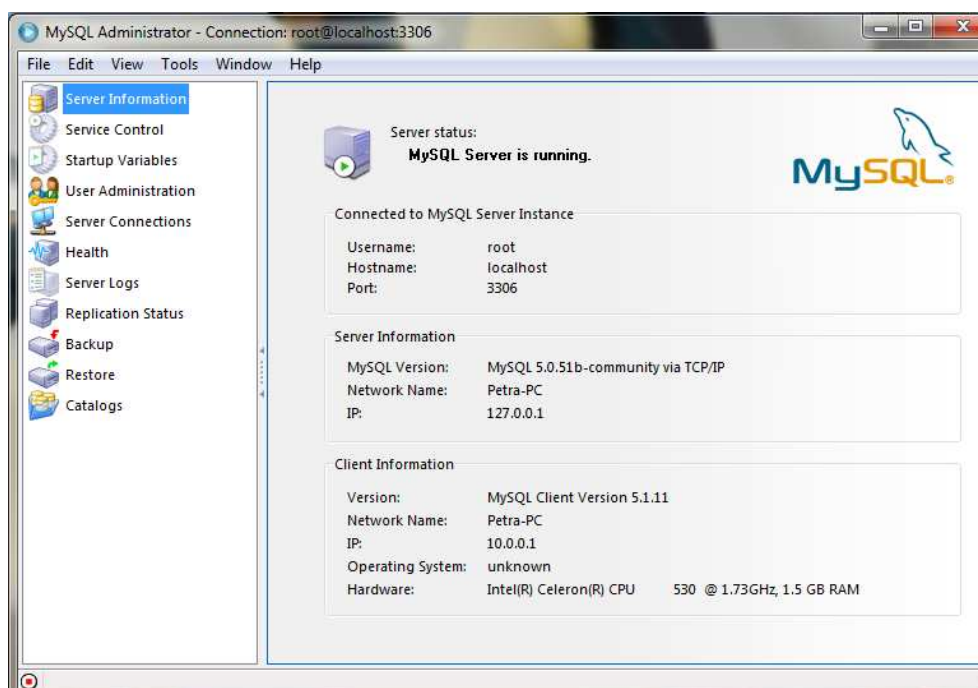
2.4 MySQL Administrator

MySQL Administrator umožňuje podrobnější administraci MySQL databáze. Například od phpMyAdmin se liší tím, že jej nemůžeme spustit z jiného počítače.

Může vykonávat tyto operace:

- spuštění a zastavení serveru
- nastavení a zjišťování aktuálního stavu serveru (obrázek 12.)
- správa uživatelů
- vytvoření a správa databází
- zálohování dat, zjišťování výkonu

Práce s programem není nijak složitá, přesto se předpokládá alespoň nějaká znalost správy MySQL.



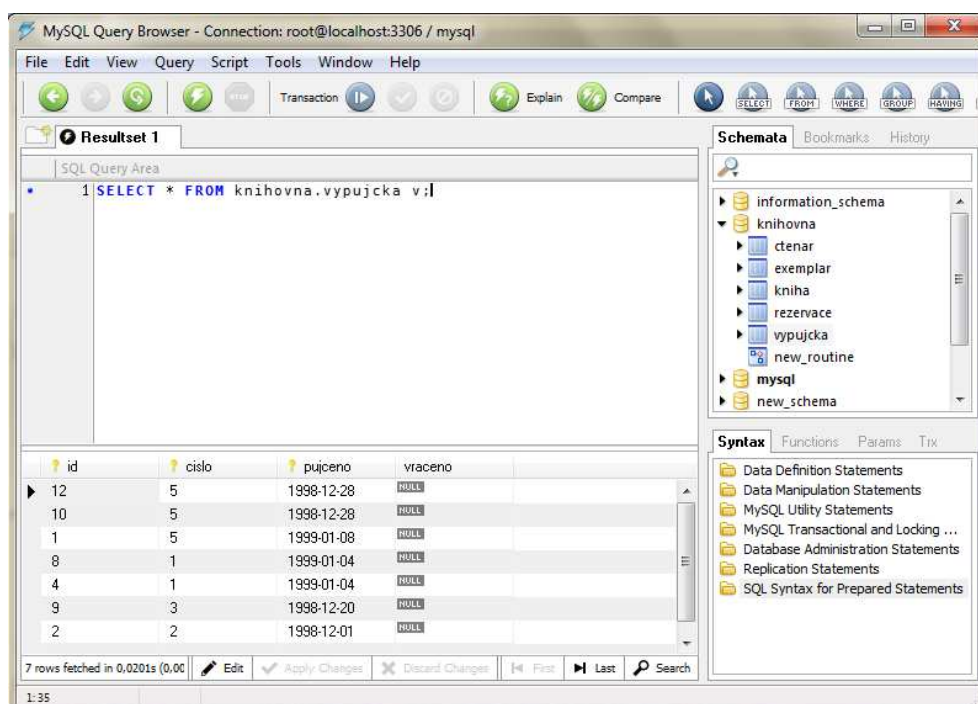
Obrázek 12. Informace o serveru

MySQL Query Browser je vizuální nástroj pro dotazování se databáze se zvýrazněním syntaxe SQL. Výsledné tabulky je přitom možno upravovat ještě před potvrzením změn v databázi. Aplikace nabízí jednoduchý přístup k mnoha funkcím, mezi které patří například vytvářet, editovat a rušit celé databáze. Obsahuje také SQL editor se zvýrazňováním syntaxe pro tvorbu dotazů nebo skriptů.

Shrnutí všech vlastností:

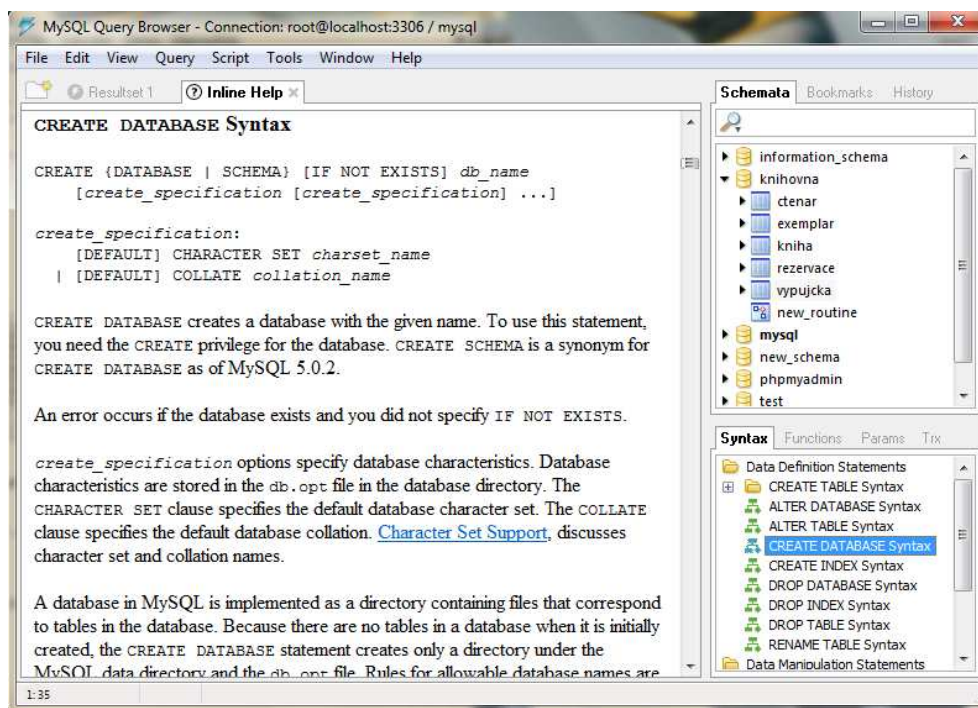
- vytvoření, editace a rušení databáze
- vytvoření tabulek a pohledů nad tabulkami
- editace dat, přidávání záznamů, správa oprávnění
- SQL editor se zvýrazňováním syntaxe
- uchovávání historie dotazů
- analyzátor tabulek, seznam procesů
- správa uživatelů

Výsledky příkazů jazyka SQL se zobrazují v tabulkách. U jednoduchých dotazů (např. `select * from tabulka`) lze měnit jednotlivé záznamy přímo v tabulce, popřípadě do tabulky přidávat nové záznamy (obrázek 13.).



Obrázek 13. Výpis tabulky vypujčka z databáze knihovna

MySQL Query Browser nabízí v pravém spodním rohu nápovědu týkající se syntaxe, funkcí a parametrů (obrázek 14.).



Obrázek 14. MySQL Query Browser nápověda

Po odzkoušení těchto tří systémů, jsem vyhodnotila, že práce s MySQL Workbench je nejjednodušší. Výhodou aplikace phpMyAdmin je, že se jedná o webový server, díky čemuž je možné ovládat MySQL odkudkoli. Na rozdíl od aplikací MySQL Workbench a MySQL Query Browser, které jsou desktopové. MySQL Query Browser nenabízí možnost vykreslení ER diagramu.

V tabulce 1. je shrnutí všech tří odzkoušených systémů.

| Funkce \ Aplikace | PhpMyAdmin | MySQL Workbench | MySQL Query Browser |
|-------------------------------|-------------------|------------------------|----------------------------|
| Editace databáze | ano | ano | ano |
| Editace tabulek | ano | ano | ano |
| SQL dotaz | ano | ano | ano |
| index, klíče | ano | ano | ano |
| Export, import | ano | ano | ano |
| Export dat do různých formátů | ano | ano | ano |
| Pohledy, procedury | ano | ano | ano |
| Historie dotazů | ano | ano | ano |
| ER diagram | ano | ano | ✗ |

Tabulka 1. Srovnání systémů

3 Navrhovaný systém

Na základě testování tří databázových správců jsem navrhla funkce informačního systému, které budu implementovat jako webovou aplikaci v PHP s podporou SŘBD MySQL. Pro práci se systémovým katalogem využiji databázi *information_schema*, ve které jsou definovány pohledy do systémového katalogu. Část této struktury je popsána dále v kapitole 4.1.

3.1 Funkce

Vyzkoušela jsem si práci se systémem phpMyAdmin, MySQL Workbench a MySQL Administrator. A po zvážení všech okolností, byly navrženy tyto funkce mého systému:

1. výpis databází a jejich vlastností
2. výpis tabulek (a pohledů) dané databáze a jejich vlastností
3. výpis atributů dané tabulky a jejich vlastností
4. výpis informací o jedné databázi ve formě vhodné pro datovou analýzu [2]
5. změna přihlašovacích údajů – možnost přihlásit se k jinému SQL serveru

Příklad datové analýzy informačního systému OBCHOD pro funkci 4.

Lineární zápis typů entit:

| | |
|----------------------------|---|
| Pobočka | (<u>číslo_pob</u> , ulice, město, stát, PSČ, telefon) |
| Prodavač | (<u>číslo_pro</u> , <u>číslo_pob</u> , poznámky, jméno, příjmení, telefon, ulice, město, stát, PSČ) |
| Zákazník | (<u>číslo_zák</u> , název_firmy, jméno, příjmení, ulice, město, stát, PSČ, telefon, <u>číslo_pob</u>) |
| Zboží | (<u>číslo_zbo</u> , text, ks_sklad, min_ks, nák_cena, pro_cena) |
| Faktura | (<u>číslo_fak</u> , <u>číslo_zák</u> , datum, cena_celkem, <i>číslo_pro</i>) |
| Položka | (<u>číslo_fak</u> , <u>číslo_řádku</u> , množství, <u>číslo_zbo</u> , pro_cena, cena_celkem_řádek) |
| (primární klíč, cizí klíč) | |

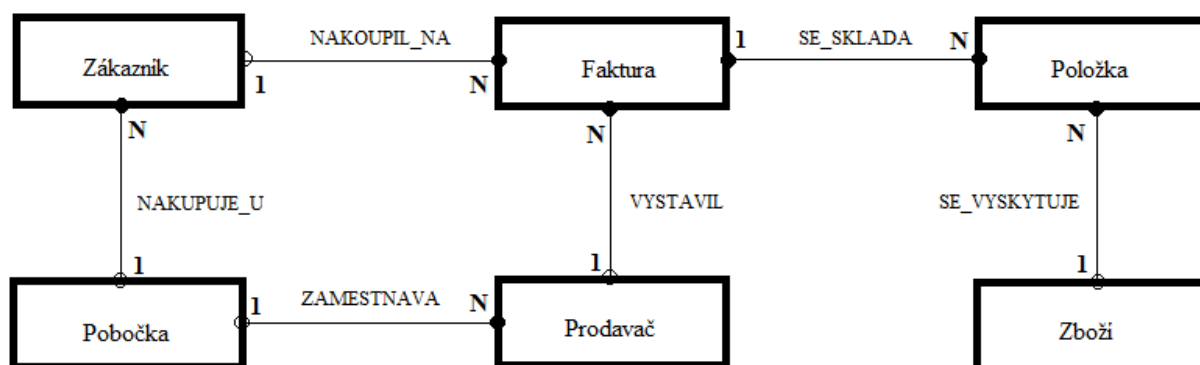
Lineární zápis typů vazeb:

| | | |
|--------------|---------------------|-----|
| SE_SKLADA | (Faktura, Položka) | 1:N |
| ZAMESTANAVA | (Pobočka, Prodavač) | 1:N |
| VYSTAVIL | (Prodavač, Faktura) | 1:N |
| NAKOUPIIL_NA | (Zákazník, Faktura) | 1:N |

SE_VYSKYTUJE (Zboží, Položka) 1:N

NAKUPUJE_U (Zákazník, Pobočka) N:1

ERD:



Obrázek 15. ER diagram databáze obchod

Datový slovník:

| Tabulka Pobočka | | | | | | |
|------------------|-----------|----------|------|------|-------|--------------------------------------|
| Atribut | Dat. Typ | Velikost | Klíč | Null | Index | Český popis |
| Číslo_pob | Character | 1 | PK | N | A | číslo pobočky 9 |
| Ulice | Character | 35 | N | N | N | ulice a číslo sídla pobočky |
| Město | Character | 15 | N | N | N | město, ve kterém pobočka sídlí |
| Stát | Character | 2 | N | N | N | stát, ve kterém pobočka sídlí |
| PSC | Character | 5 | N | N | N | PSC pobočky, 99999 |
| Telefon | Character | 12 | N | N | N | telefonní číslo pobočky |
| Tabulka Prodavač | | | | | | |
| číslo_pro | Character | 3 | PK | N | A | číslo prodavače, 999 |
| číslo_pob | Character | 1 | FK | N | N | číslo pobočky 9, cizí klíč z Pobočka |
| poznámky | Character | 200 | N | A | N | poznámka |
| Jméno | Character | 30 | N | N | A | jméno prodavače |
| příjmení | Character | 50 | N | N | A | příjmení prodavače |

Tabulka 2. Část datového slovníku databáze obchod

4 Datová analýza

Mezi nejdůležitější částí návrhu databázového systému je volba vhodného uložení dat, definice tabulek a jejich položek včetně definice vazebních vztahů mezi nimi. Proces jejich návrhu označujeme pojmem datová analýza [1].

Výsledný konceptuální model obsahuje:

- Lineární zápis seznamu typu entit a jejich atributů
- ER diagram
- Datový slovník + IO

4.1 Lineární zápis

SŘBD MySQL si informace o objektech v databázi udržuje v systémovém katalogu. Přístup na čtení do toho systémového katalogu je možný přes databázi *information_schema*. Lineární textový zápis popisuje některé entity z tohoto schématu a vazby:

SCHEMATA (CATALOG_NAME, SCHEMA_NAME, DEFAULT_CHARACTER_SET_NAME, DEFAULT_COLLATION_NAME, SQL_PATH)

TABLES (TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, TABLE_TYPE, ENGINE, VERSION, ROW_FORMAT, TABLE_ROWS, AVG_ROW_LENGTH, DATA_LENGTH, DATA_FREE, AUTO_INCREMENT, CREATE_TIME, UPDATE_TIME, CHECK_TIME, TABLE_COLLATION, CHECKSUM, CREATE_OPTIONS, TABLE_COMMENT)

COLUMNS (TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, ORDINAL_POSITION, COLUMN_DEFAULT, IS_NULLABLE, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH, CHARACTER_OCTET_LENGTH, NUMERIC_PRECISION, NUMERIC_SCALE, CHARACTER_SET_NAME, COLLATION_NAME, COLUMN_TYPE, COLUMN_KEY, EXTRA, PRIVILEGES, COLUMN_COMMENT)

KEY_COLUMN_USAGE (CONSTRAINT_NAME, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, REFERENCED_TABLE_SCHEMA, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME, CONSTRAINT_CATALOG, CONSTRAINT_SCHEMA, TABLE_CATALOG, ORDINAL_POSITION, POSITION_IN_UNIQUE_CONSTRAINT)

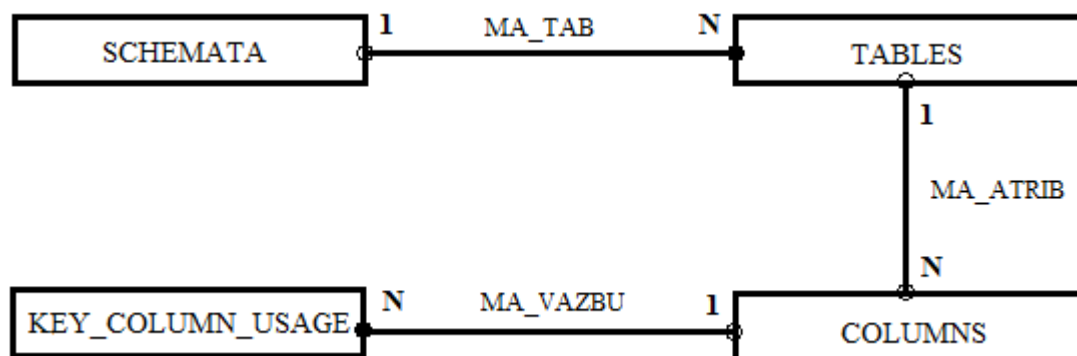
4.2 Vazby

Vazby v katalogu nejsou zachyceny na úrovni jazyka SQL v příkazu `create table`, ale při prohlížení dat v katalogu, hodnoty atributů `schema_name` v tabulce `schemata` odpovídaly hodnotám atributů `table_schema` v tabulce `tables`. Mezi tabulkami `tables` a `columns` je vazba tvořena stejnojmennými atributy `table_name`. Na základě těchto úvah, můžeme nakreslit následující ER diagram.

4.3 ER diagram

ER diagram graficky znázorňuje objekty a vztahy formou grafu, kde

- obdélník představuje entitní typ
- spojovací čára označuje typ vztahu mezi entitními typy (obrázek 16.)



Obrázek 16. ER diagram

4.4 Datový slovník + integritní omezení

Aby typy entit byly zadány úplně, musíme ke každému atributu určit ještě několik podrobnějších údajů. Vše máme uspořádáno v tabulce atributů – v tzv. datovém slovníku. Integritní omezení (IO) jsou logická omezení na typy a hodnoty atributů, entit a vazeb tak, aby schéma konceptuální co nejpřesněji odpovídalo zobrazované realitě. Zadáávají se graficky v ERD nebo popisem v datovém slovníku nebo doplňujícím textem formou poznámky za datovým slovníkem.

V následujících tabulkách jsou popsány atributy tří entitních typů, které využívám ve své aplikaci. Systémový katalog *information_schema* obsahuje více tabulek, než jsem využila.

SCHEMATA

| Sloupec | Typ | Nulový | Výchozí | Odkazuje na |
|----------------------------|---------------|--------|---------|--|
| CATALOG_NAME | varchar (512) | Ano | NULL | |
| SCHEMA_NAME | varchar (64) | Ne | | |
| DEFAULT_CHARACTER_SET_NAME | varchar (64) | Ne | | CHARACTER_SETS -> CHARACTER_SET_NAME |
| DEFAULT_COLLATION_NAME | varchar (64) | Ne | | |
| SQL_PATH | varchar (512) | Ano | NULL | |

Tabulka 3. Tabulka schemata – seznam databází v SŘBD

TABLES

| Sloupec | Typ | Nulový | Výchozí | Odkazuje na |
|-----------------|---------------|--------|---------|---------------------------------|
| TABLE_CATALOG | varchar (512) | Ano | NULL | |
| TABLE_SCHEMA | varchar (64) | Ne | | SCHEMATA -> SCHEMA_NAME |
| TABLE_NAME | varchar (64) | Ne | | |
| TABLE_TYPE | varchar (64) | Ne | | |
| ENGINE | varchar (64) | Ano | NULL | |
| VERSION | bigint (21) | Ano | NULL | |
| ROW_FORMAT | varchar (10) | Ano | NULL | |
| TABLE_ROWS | bigint (21) | Ano | NULL | |
| AVG_ROW_LENGTH | bigint (21) | Ano | NULL | |
| DATA_LENGTH | bigint (21) | Ano | NULL | |
| MAX_DATA_LENGTH | bigint (21) | Ano | NULL | |
| INDEX_LENGTH | bigint (21) | Ano | NULL | |
| DATA_FREE | bigint (21) | Ano | NULL | |
| AUTO_INCREMENT | bigint (21) | Ano | NULL | |
| CREATE_TIME | datetime | Ano | NULL | |
| UPDATE_TIME | datetime | Ano | NULL | |
| CHECK_TIME | datetime | Ano | NULL | |
| TABLE_COLLATION | varchar (64) | Ano | NULL | COLLATIONS -> COLLATION_NAME |
| CHECKSUM | bigint (21) | Ano | NULL | |
| CREATE_OPTIONS | varchar (255) | Ano | NULL | |
| TABLE_COMMENT | varchar(80) | Ne | | |

Tabulka 4. Tabulka tables

COLUMNS

| Sloupec | Typ | Nulový | Výchozí | Odkazuje na |
|--------------------------|---------------|--------|---------|---|
| TABLE_CATALOG | varchar (512) | Ano | NULL | |
| TABLE_SCHEMA | varchar (64) | Ne | | SCHEMATA -> SCHEMA_NAME |
| TABLE_NAME | varchar (64) | Ne | | |
| COLUMN_NAME | varchar (64) | Ne | | |
| ORDINAL_POSITION | bigint (21) | Ne | | |
| COLUMN_DEFAULT | longtext | Ano | NULL | |
| IS_NULLABLE | varchar (3) | Ne | | |
| DATA_TYPE | varchar (64) | Ne | | |
| CHARACTER_MAXIMUM_LENGTH | bigint (21) | Ano | NULL | |
| CHARACTER_OCTET_LENGTH | bigint (21) | Ano | NULL | |
| NUMERIC_PRECISION | bigint (21) | Ano | NULL | |
| NUMERIC_SCALE | bigint (21) | Ano | NULL | |
| CHARACTER_SET_NAME | varchar (64) | Ano | NULL | CHARACTER_SETS -> CHARACTER_SET_NAME |
| COLLATION_NAME | varchar (64) | Ano | NULL | COLLATIONS -> COLLATION_NAME |
| COLUMN_TYPE | longtext | Ne | | |
| COLUMN_KEY | varchar (3) | Ne | | |
| EXTRA | varchar (20) | Ne | | |
| PRIVILEGES | varchar (80) | Ne | | |

Tabulka 5. Tabulka columns

5 Implementace systému

Pro implementaci mého informačního systému jsem zvolila webovou aplikaci pomocí jazyka PHP s použitím SŘBD MySQL na serveru a libovolného prohlížeče na klientovi. Vzhledem k tomuto vybranému softwaru jsem si nainstalovala na můj notebook do OS Windows produkt VertrigoServ, kde součástí instalace je PHP, MySQL, Apache, phpMyAdmin a další programy [9].

Výsledkem by měl být funkční systém, který bude obsahovat mnou navržené funkce. Tento systém je konkrétně implementován kombinací jazyků PHP a HTML s přístupem na MySQL server. Jako vývojové prostředí byl zvolen editor PS Pad. Vytvořila jsem 12 zdrojových souborů:

- Db.php – program pro načtení a výpis seznamu databází
- Tb.php – program pro načtení a výpis seznamu tabulek z vybrané databáze
- Cl.php – program pro načtení a výpis seznamu atributů jedné tabulky
- Fce.php – zdrojový kód opakovaně používaných funkcí
- Zmena.php – zdrojový kód pro změnu parametrů připojení (server, login, heslo)
- Vypis.php – zdrojový kód pro výpis informací o databázi v požadovaném tvaru datové analýzy (lineární zápis, seznam typů vazeb, datový slovník)
- Session_setting.php – zdrojový kód pro vytvoření a nastavení session proměnných [3]
- Styly.css – soubor stylů pro pěkný vzhled stránek
- Erd.php – program pro načtení informací o tabulkách a vazbách v databázi a vygenerování příkazů pro MetaUML
- Erd.tex – zdrojový soubor pro vysázení ER diagramu
- Pok1.mp – soubor s příkazy jazyka MetaUML
- Pok1.1 – vygenerovaný soubor s příkazy pro vykreslení obrázku ER diagramu

Jelikož protokol HTTP je bezstavový, bylo potřeba vymyslet, jak si zapamatovat hodnoty některých proměnných mezi jednotlivými stránkami. Jako ideální se nabízely session proměnné.

5.1 PHP

Jazyk PHP je skriptovací jazyk vykonávaný na straně serveru vkládaný do běžného HTML kódu. To znamená, že každou stránku, která obsahuje PHP skripty, server nejprve vezme a vykoná všechny příkazy v PHP, které jsou ve stránce uvedené, poté pošle klientovi již čistý HTML kód, který je výsledkem běhu skriptu. Server může PHP skripty teoreticky hledat ve všech odesílaných souborech, ale zpravidla je nakonfigurován tak, aby je hledal v souborech s příponami .php, .php3 nebo .phtml. Příkazy PHP jsou vkládány přímo do HTML kódu a jsou od něj odděleny tagy `<? a ?>` (nebo `<?php a ?>`).

Data si můžeme ukládat buď do obyčejných textových souborů nebo do databáze (PHP si dobře rozumí s většinou běžně používaných databází např. MySQL). Velmi snadné je zpracovávání dat z formulářů, snadno vytvoříte různé on-line testy včetně statistik úspěšnosti dosavadních návštěvníků nebo naprogramujete kvalitní reklamní systém.

5.1.1 Historie PHP

Počátky PHP sahají do roku 1994, kdy Rasmus Lerdorf vytvořil jednoduchý systém evidence přístupů ke svému webu, nejdříve v jazyce Perl, poté v C. Ten se rozšířil mezi další uživatele, kteří přicházeli s požadavky na vylepšení. Vznikl tak systém Personal Home Page Tools, později Personal Home Page Construction Kit.

- PHP/FI

Rasmus Lerdorf vytvořil i nástroj umožňující začleňování SQL dotazů a tím zpřístupnění databází na serveru – Forms Interpreter (FI). V roce 1997 bylo PHP/FI 2.0 oficiálně uvolněno, brzy potom vznikla verze PHP 3.0.

- PHP 3

PHP 3.0 vytvořil Andi Gutmans a Zeev Suraski. Původní zkratka dostává nový význam – PHP: Hypertext Preprocessor.

- PHP 4

PHP 4.0, oficiálně uvolněné v roce 2000 je výkonnější verzí než PHP 3, přidává například podporu pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a nové jazykové konstrukty.

5.1.2 Psaní PHP skriptů

K programování v PHP budeme potřebovat PHP, Apache, editor, PHP manuál, pro práci s databázemi i MySQL. Vše jmenované obsahuje balíček VertrigoServ, který je volně stažitelné z internetu. Pro psaní PHP skriptů si vystačíme s obyčejným editorem (např. PSPad).

Důležitou součástí při práci jsou také manuály, které nám umožní nápovědu při problémech [4].

5.1.3 Ukázky PHP kódu

Připojení k databázi

Hodnoty nutné pro připojení k databázi jsou uloženy v session proměnných, při změně těchto hodnot funkce připojení načte tyto změněné hodnoty ze session proměnných.


```

14 function pripojeni(){
15 @$spojeni = mysql_connect($_SESSION["server"], $_SESSION["login"], $_SESSION["password"]);
16     if (!$spojeni) echo '<a href="zmena.php">Zmena</a>';
17 mysql_select_db("information_schema")
18     or die("nepodařilo se připojit k db");
19 }

```

Obrázek 17. Připojení k databázi MySQL

Automatizace načítání a výpis dat z katalogu

Názvy atributu které jsou nutné pro výpis, jsou uloženy v pomocných polích, které využívám při výpisu hlaviček jednotlivých sloupců a při generování SQL dotazů.

```
$cz_schemata = Array("DATABÁZE", "KÓDOVÁNÍ", "POROVNÁNÍ");
```

```
$cz_tables = Array ("TABULKA", "TYP", "ZPRACOVATEL", "VERZE", "FORMÁT ŘÁDKU",
"PRŮM.DÉLKA ŘÁDKU", "AUTOMATIC.ČÍSLOVÁNÍ", "ČAS KONTROLY", "POROVNÁNÍ
TABULKY", "PARAMETRY VYTVOŘENÍ");
```

```
$cz_columns = Array ("KATALOG", "DATABÁZE", "TABULKA", "SLOUPEC", "POŘADOVÉ
UMÍSTĚNÍ", "VÝCHOZÍ SLOUPEC", "IS NULL", "DATOVÝ TYP", "MAX DÉLKA", "DÉLKA
ŘETĚZCE V BYTECH", "PŘESNOST", "POČET DESET.MÍST", "KÓDOVÁNÍ", "POROVNÁNÍ",
"TYP SLOUPCE", "KLÍČ", "EXTRA", "PRIVILEGIA", "KOMENTÁŘ");
```

```
$cz_key_columns_Z = Array ("SLOUPEC2", "DATABÁZE", "TABULKA", "SLOUPEC");
```

```
$cz_key_columns_H = Array ("SLOUPEC1", "DATABÁZE2", "TABULKA2", "SLOUPEC2");
```

```
$cz_vazby = Array ("NÁZEV VAZBY", "TABULKA1", "SLOUPEC1", "TABULKA2",
"SLOUPEC2");
```

Příklad výpisu hlavičky atributů seznamu databází s použitím pole *schemata*.

```

37 <!-- Výpis atributů -->
38
39 <h1 style="color: slateblue">Výpis atributů z tabulky <?echo $tb?></h1>
40 <table border="1">
41 <br>
42 <tr><td>#</td>
43 <? for ($i=0; $i<count($columns); $i++) { //hlavička tabulky
44     echo "<td>".$cz_columns[$i]."</td>";
45 } ?>

```

Obrázek 18. Hlavička tabulky pro výpis atributů

Zdrojový kód pro načtení hodnot parametrů předaných z volajícího programu, vygenerování seznamu atributů z hodnot uložených v poli *tables* a jeho použití za klíčovým slovem *select* v SQL dotazu.

```

12 $tb = $_REQUEST["tab"];
13 $db = $_REQUEST["schema"];
14
15 $sez_atr='';
16 for ($i=0;$i<count($tables);$i++) {           //seznam atributů
17     if($i==count($tables)-1)
18         $sez_atr .=$tables[$i];
19     else $sez_atr .=$tables[$i].", ";
20 }
21 $select_ret="select ".$sez_atr." from tables where table_schema='$db'";
22
23 //echo $select_ret;
24 $vysledek = mysql_query($select_ret);

```

Obrázek 19. Kód pro načtení hodnot

Část zdrojového kódu pro potvrzení primárního klíče, kurzívou vypsání cizího klíče a výpis ostatních atributů.

```

64 if ($zaznaml->column_key == "PRI") echo "<u>";
65 $col = $zaznaml->sloupec;
66 $dotazFK = "select count(*)
67 from key_column_usage u
68 where u.referenced_table_name is not null and
69       u.table_schema='$db' and
70       u.table_name = '$tab' and
71       u.column_name = '$col' ";
72 // echo "<!--".$dotazFK." -->";
73 $cizi = mysql_query($dotazFK);
74 $pocet = mysql_fetch_row($cizi);
75 if ($pocet[0] >0) echo "<i>";
76 echo $zaznaml->sloupec;
77 if ($pocet[0] >0) echo "</i>";
78 if ($zaznaml->column_key == "PRI") echo "</u>";

```

Obrázek 20. Kód pro vypsání podtrženého textu a kurzívou

5.2 HTML

HTML znamená HyperText Markup Language = hypertextový značkovací jazyk. S jeho pomocí tvoříme webové stránky. První definici jazyka vytvořil v roce 1991 Tim Berners-Lee.

HTML kód je kombinací normálního textu a značek. Smysl značek je v oddělování a vyznačování částí textu. Těmto značkám se také říká tagy.

5.2.1 Syntaxe jazyka HTML

Nejdůležitější pravidla pro psaní zdrojového kódu v HTML jazyce:

- Nezáleží na velikosti písma, př.: <head> = <HEAD>
- V názvech souboru a v adresách záleží na velikosti písma
- Jedna mezera, konec řádku, tabulátor jsou brány jako jedna mezera
- X mezer, x prázdných řádků jsou také brány jako jedna mezera
- Tagy, které prohlížeč nezná, jako by nebyly
- Na začátku tagu nesmí být mezera, př.: < head> je špatně
- Použití uvozovek u hodnot atributů je nutné, jen pokud hodnota atributů obsahuje mezeru, př.:
- Speciální znaky typu © se do zdrojového kódu zapisují jako posloupnost znaků, př.: \$nbsp (pevná mezera)
- Poznámky se vkládají mezi značky <! ... >, tato poznámka se nezobrazí

Nejčastěji používaná struktura HTML souboru[6]:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1250">
  <title>Titulek stránky</title>
  <link rel="stylesheet" href="style/muj-styl.css" type="text/css">
</head>
<body>
... zde pokračuje dokument
```

Obrázek 21. Příklad hlavičky HTML

V prvním řádku je definován typ dokumentu. V části <head> jsou definovány doplňkové vlastnosti dokumentu, v části <body> je samotný dokument.

5.3 CSS

Cascading Style Sheets[11] neboli kaskádové styly nám určují grafické vlastnosti webových stránek. Například si můžeme nastavit barvu písma, velikost písma, pozadí, zarovnání a mnoho dalších. CSS formátuje obsah HTML, XHTML a XML dokumentů.

Připojení externího dokumentu style.css:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Část ukázky kaskádového stylu:

```
body {  
    margin:0px 0;  
    padding:0;  
    background:url(bodybg.gif) repeat;  
    font: 74% Arial, Sans-Serif;  
    color:#ccc;  
    line-height: 1.4em;  
}  
  
.content {  
    background: #101010;  
    color:#a3b2ba;  
    margin: 0 auto;  
    padding: 0;  
    width: 800px;  
    border-left:6px solid #000;  
    border-right:6px solid #000;  
}
```

Obrázek 22. Část kódu CSS

5.4 Generování ER diagramu do obrázku

Pro vygenerování ER diagramu z databáze budu potřebovat informace z databáze *information_schema* z pohledů *tables*, *columns*, *key_column_usage*. Z pohledů *tables* získám názvy tabulek pro vykreslení typů entit, z pohledů *columns*, *key_column_usage* získám informace pro vykreslení typů vazeb mezi entitami. Vazby typu primární, cizí klíč je uložena v pohledu *key_column_usage* a v pohledu *columns* je nastavena hodnota atributu *column_key* = MUL.

Pro vygenerování ER diagramu použiji knihovnu MetaUML systému TeX Live[12]. Na následujícím výpisu vidíte vygenerovaný zdrojový kód pro vykreslení části třídního diagramu v jazyce MetaUML. Tento zdrojový kód je vygenerován programem *erd.php*, který načte informace z databáze *information_schema* a z pohledů uvedených v prvním odstavci.

```

input metauml;
beginfig(1);Class.A("ctenar") ();
Class.B("exemplar") ();
Class.C("kniha") ();
Class.D("rezervace") ();
Class.E("vypujcka") ();
A.nw=(0,0);
leftToRight(20)(A,B,C,D,E);topToBottom.left(20)(A);
drawObjects(A,B,C,D,E);
link(association)(pathStepY(B.s,C.s,-12));
link(association)(pathStepY(D.s,C.s,-10));
link(association)(pathStepY(D.s,A.s,-8));
link(association)(pathStepY(E.s,A.s,-6));
link(association)(pathStepY(E.s,B.s,-4));
endfig;
end;|

```

Obrázek 23. Zdrojový kód databáze knihovna pok1.mp

Jednotlivé soubory *.mp, je potřeba příkazem `mpost *.mp` přeložit do souboru *.1, které se vkládají pomocí příkazu `\includegraphics{pok.1}` do patřičného souboru tex.

```

\documentclass{article}
\usepackage[cp1250]{inputenc}

\usepackage{graphicx}
\DeclareGraphicsRule{*}{mps}{*}{}

\date{13.4.2011}
\begin{document}

\section{ER diagram}
\includegraphics{pok1.1}
\end{document}

```

Obrázek 24. Erd.tex

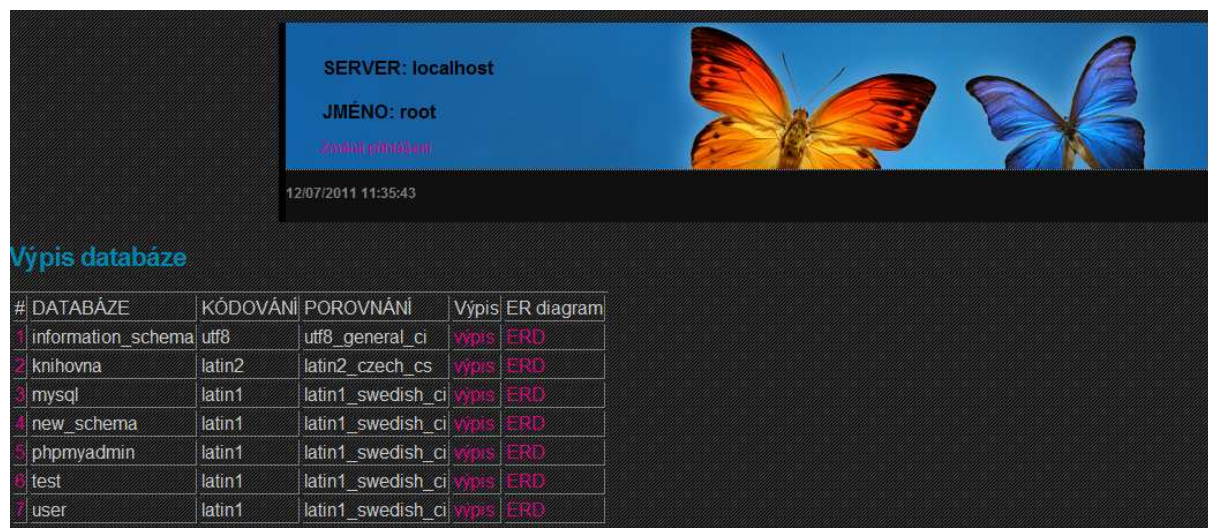
6 Provoz

Vytvořený systém *Model databáze* byl testován na `localhostu` v operačním systému Windows 7. V internetovém prohlížeči Google Chrome. Následující kapitola obsahuje obrázky a popis běhu programu.

6.1 Vzhled systému

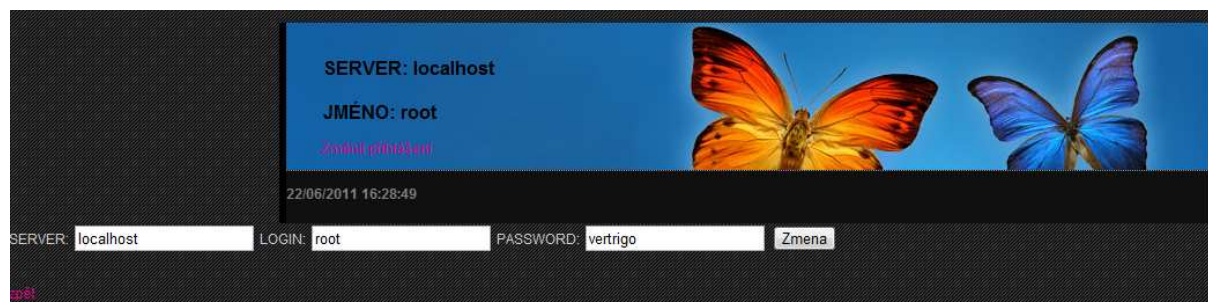
Na obrázku 25. je zobrazena úvodní stránka po spuštění systému. Podrobnější popis je zaznamenán v uživatelské příručce. Na této stránce vidíme seznam všech dostupných databází.

Automatické přihlášení uživatele je jako `root`.



Obrázek 25. Výpis seznamu databází


Pro změnu uživatele nám slouží formulář, který vidíme na obrázku 26., kde uživatel může zadat jméno jiného serveru, existující login a heslo uživatele a poté po úspěšném přihlášení uvidí výpis databází ze zadaného serveru.



Obrázek 26. Formulář pro přihlášení

Na obrázku 27. vidíme výpis všech tabulek a jejich vlastností z databáze knihovna.

SERVER: localhost
JMÉNO: root
Zobrazit přehled



11/07/2011 23:51:20

Výpis tabulek z databáze knihovna


| # | TABULKA | TYP | ZPRACOVATEL | VERZE | FORMÁT ŘÁDKU | PRŮM.DĚLKA ŘÁDKU | AUTOMATIC.ČÍSLOVÁNÍ | ČAS KONTROLY | POROVNÁNÍ TABULKY |
|---|-----------|---------------|-------------|-------|-----------------|---------------------|---------------------|-----------------|----------------------|
| 1 | ctenar | BASE TABLE | InnoDB | 10 | Compact | 3276 | 6 | | latin2_czech_cs |
| 2 | exemplar | BASE TABLE | InnoDB | 10 | Compact | 0 | 1 | | latin2_czech_cs |
| 3 | kniha | BASE TABLE | InnoDB | 10 | Compact | 2048 | | | latin2_czech_cs |
| 4 | rezervace | BASE TABLE | InnoDB | 10 | Compact | 3276 | | | latin2_czech_cs |
| 5 | vypujcka | BASE TABLE | InnoDB | 10 | Compact | 2340 | | | latin2_czech_cs |

[Zpět](#)

Obrázek 27. Výpis tabulek databáze knihovna

Obrázek 28. zobrazuje výpis atributů z tabulky *exemplar* a také výpis tabulek závislých na hlavní tabulce a tabulek závislých na ni.

SERVER: localhost
JMÉNO: root
Zobrazit přehled



11/07/2011 23:52:55

Výpis atributů z tabulky exemplar

| # | KATALOG | DATABÁZE | TABULKA | SLOUPEC | POŘADOVÉ UMÍSTĚNÍ | VÝCHOZÍ SLOUPEC | IS NULL | DATOVÝ TYP | MAX DĚLKA | DĚLKA ŘETĚZCE V BYTECH | PŘESNOST | POČET DESET.MÍST | KÓDOVÁNÍ | POROVNÁNÍ |
|---|---------|----------|----------|---------|----------------------|--------------------|------------|---------------|--------------|---------------------------------|----------|---------------------|----------|-----------------|
| 1 | | knihovna | exemplar | id | 1 | | NO | int | | | 10 | 0 | | |
| 2 | | knihovna | exemplar | isbn | 2 | | NO | varchar | 16 | 16 | | | latin2 | latin2_czech_cs |
| 3 | | knihovna | exemplar | koupeno | 3 | 0000-00-00 | YES | date | | | | | | |
| 4 | | knihovna | exemplar | cena | 4 | | YES | decimal | | | 6 | 0 | | |

Tabulka je závislá na: (hlavní tabulky)

| # | SLOUPEC1 | DATABÁZE2 | TABULKA2 | SLOUPEC2 |
|---|----------|-----------|----------|----------|
| 1 | isbn | knihovna | kniha | isbn |

Jiné závislé tabulky na ni

| # | SLOUPEC2 | DATABÁZE | TABULKA | SLOUPEC |
|---|----------|----------|---------|---------|
|---|----------|----------|---------|---------|

Obrázek 28. Výpis atributů z tabulky exemplar a tabulky závislé

Na obrázku 29. jsou zobrazeny lineární zápis typu entit a typu vazeb.

SERVER: localhost

JMÉNO: root

[Znovit přihlášení](#)

11/07/2011 23:54:09

Datová analýza - databáze knihovna

Lineární zápis typů entit

1 **ctenar**(cislo, jmeno, prijmeni, ulice, mesto, psc, rc, telefon, vzdelani)

2 **exemplar**(id, isbn, koupeno, cena)

3 **kniha**(isbn, nazev, autor, zeme)

4 **rezervace**(isbn, cislo, rezervovano)

5 **vypujcka**(id, cislo, pujceno, vraceno)

Pozn.: (Primární klíč, Cizí klíč)

Lineární zápis typů vazeb

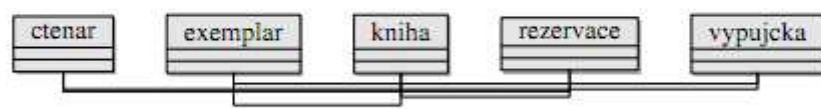
| # | NÁZEV VAZBY | TABULKA1 | SLOUPEC1 | TABULKA2 | SLOUPEC2 |
|---|-----------------|----------|----------|----------|----------|
| 1 | EXEMPL_KNIHA_FK | exemplar | isbn | kniha | isbn |

Obrázek 29. Lineární zápis typu entit a vazeb databáze knihovna

| Výpis atributů | | | | | | | | | | | | | | |
|----------------|---------|----------|----------|----------|-------------------|-----------------|---------|------------|-----------|------------------------|----------|------------------|----------|-----------------|
| # | KATALOG | DATABÁZE | TABULKA | SLOUPEC | POŘADOVÉ UMÍSTĚNÍ | VÝCHOZÍ SLOUPEC | IS NULL | DATOVÝ TYP | MAX DÉLKA | DÉLKA ŘETĚZCE V BYTECH | PŘESNOST | POČET DESET MÍST | KÓDOVÁNÍ | POROVNÁNÍ |
| 1 | | knihovna | ctenar | cislo | 1 | | NO | int | | | 10 | 0 | | |
| 2 | | knihovna | ctenar | jmeno | 2 | | NO | varchar | 10 | 10 | | | latin2 | latin2_czech_cs |
| 3 | | knihovna | ctenar | prijmeni | 3 | | NO | varchar | 15 | 15 | | | latin2 | latin2_czech_cs |
| 4 | | knihovna | ctenar | ulice | 4 | | YES | varchar | 20 | 20 | | | latin2 | latin2_czech_cs |
| 5 | | knihovna | ctenar | mesto | 5 | | YES | varchar | 15 | 15 | | | latin2 | latin2_czech_cs |
| 6 | | knihovna | ctenar | psc | 6 | | YES | char | 5 | 5 | | | latin2 | latin2_czech_cs |
| 7 | | knihovna | ctenar | rc | 7 | | YES | char | 10 | 10 | | | latin2 | latin2_czech_cs |
| 8 | | knihovna | ctenar | telefon | 8 | | YES | varchar | 15 | 15 | | | latin2 | latin2_czech_cs |
| 9 | | knihovna | ctenar | vzdelani | 9 | | YES | char | 1 | 1 | | | latin2 | latin2_czech_cs |
| 10 | | knihovna | exemplar | id | 1 | | NO | int | | | 10 | 0 | | |
| 11 | | knihovna | exemplar | isbn | 2 | | NO | varchar | 16 | 16 | | | latin2 | latin2_czech_cs |
| 12 | | knihovna | exemplar | koupeno | 3 | 0000-00-00 | YES | date | | | | | | |
| 13 | | knihovna | exemplar | cena | 4 | | YES | decimal | | | 6 | 0 | | |
| 14 | | knihovna | kniha | isbn | 1 | | NO | varchar | 16 | 16 | | | latin2 | latin2_czech_cs |
| 15 | | knihovna | kniha | nazev | 2 | | NO | varchar | 32 | 32 | | | latin2 | latin2_czech_cs |
| 16 | | knihovna | kniha | autor | 3 | | YES | varchar | 24 | 24 | | | latin2 | latin2_czech_cs |
| 17 | | knihovna | kniha | zeme | 4 | CZ | YES | char | 2 | 2 | | | latin2 | latin2_czech_cs |

Obrázek 30. Část datového slovníku

1 ER diagram



Obrázek 31. Vykreslení ER diagramu

7 Závěr

Cílem této bakalářské práce bylo seznámit se s problematikou již existujících správců databázových systémů a vybrat nejpoužívanější funkce a porovnat je vzhledem k funkcím, již používaným. Na základě tohoto průzkumu poté vytvořit vlastní verzi systému pro generování modelu databáze včetně vykreslení ER diagramu.

K porovnání byly vybrány systémy phpMyAdmin, MySQL Workbench a MySQL Administrator. Byly vyzkoušeny všechny funkce těchto aplikací a výsledek byl uveden ve srovnávací tabulce. V dalších krocích byly vybrány funkce vhodné pro vytvoření vlastního systému. Po vytvoření datové analýzy byl tento systém implementován jako webová aplikace v jazycích PHP a HTML, doplněný o CSS styly. Ke spuštění systému je potřeba libovolný prohlížeč na klientovi a web server s PHP a SŘBD MySQL. V systémovém katalogu se mi podařilo najít způsob, jak jsou uloženy vazby mezi tabulkami a proto vygenerovaný ER diagram obsahuje jak typy entit, tak typy vazeb mezi entitami. K vytvoření ER diagramu byl použit můj program erd.php, který z načteného schématu databáze vygeneroval příkazy v jazyce vhodném pro MetaUML. Pomocí sázecího program TeX (a jeho balíčku MetaUML), na základě mnou vytvořených vstupních dat, byl vygenerován ER diagram do pdf souboru. Vzhledem k bodům zadání lze konstatovat, že cíle práce byly splněny.

Tento systém pro generování modelu databáze může být užitečným základem pro další vývoj a rozšíření této aplikace o další funkce, případně jiné SŘBD.

Literatura

1. ŠARMANOVÁ, Jana. *Teorie zpracování dat* [online]. druhé, přepracované. Ostrava: VŠB – Technická univerzita Ostrava, 2007 [cit. 2010-11-09]. Dostupné z WWW: <http://www.elearn.vsb.cz/archivcd/FEI/TZD/TZD.pdf>. ISBN 978-80-248-1498-8.
2. ŠARMANOVÁ, Jana. *Databázové a informační systémy* [online]. první. Ostrava : VŠB – Technická univerzita Ostrava, 2007 [cit. 2010-11-09]. Dostupné z WWW: <http://www.elearn.vsb.cz/archivcd/FEI/DAIS/DAIS.pdf>. ISBN 978-80-248-1499-5.
3. KOSEK, Jiří. *PHP Tvorba interaktivních internetových aplikací : Podrobný průvodce*. Vydání 1. Praha : Grada Publishing, 1999. 492 s. ISBN 80-7169-373-1.
4. *Php* [online]. 2001 [cit. 2011-04-28]. Dostupné z WWW: <http://cz.php.net/>.
5. *MySQL* [online]. 2000 [cit. 2011-04-30]. MySQL Documentation: MySQL Reference Manuals. Dostupné z WWW: <http://dev.mysql.com/doc/>.
6. *Jak psát web* [online]. 2007 [cit. 2011-04-24]. Syntaxe HTML. Dostupné z WWW: <http://www.jakpsatweb.cz/html/syntaxe.html>.
7. *Interval.cz* [online]. 2009 [cit. 2011-04-24]. PHP - základní informace. Dostupné z WWW: <http://interval.cz/clanky/php-zakladni-informace/>.
8. *Databázový svět* [online]. 2004 [cit. 2011-04-24]. Database Workbench 3. Dostupné z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2008010702>.
9. *EDownload* [online]. 2000 [cit. 2011-04-30]. VertrigoServ. Dostupné z WWW: <http://www.edownload.cz/sw/vertrigoserv/>.
10. *ALTOVA* [online]. 2005 [cit. 2011-05-04]. DatabaseSpy Graphical Database Design Tool. Dostupné z WWW: <http://www.altova.com/databasespy/database-design.html>.
11. *Free CSS Templates, CSS Layouts & More!* [online]. 2007-2011 [cit. 2011-07-12]. Dostupné z WWW: <http://www.free-css.com>.
12. *TeX Live* [online]. 5.7.2011 [cit. 2011-07-12]. TeX Live - TeX Users Group. Dostupné z WWW: <http://tug.org/texlive/>.

Seznam příloh

Příloha č. 1: Uživatelská příručka

Příloha č. 2: Programátorská příručka

Adresářová struktura přiloženého DVD

| | |
|-----------|---|
| /Aplikace | Zdrojové soubory k informačnímu systému <i>Model databáze</i> |
| /Prirucky | Uživatelská a programátorská příručka k aplikaci |
| /Texty | Soubory s textem práce, zadání, klíčová slova a abstrakt |